1. [Marks 1]

Write the definition of a function $f(n) \in \Omega(g(n))$.

If $f(n) \in \Omega(g(n))$ (is true iff) $f(n) \geq c \cdot g(n)$ (and there exists) $\forall n \geq n_0$

2. [Marks 2]

What is Θ, C_1, C_2 and n_0 for the equation $f(n) = n^3 - 10n^2 - 2n + 1$.

Big Oh: $f(n) \leq n^3 + 10n^3 + 2n^3 + 1n^3$

$O(n^3)$: $\leq 14n^3$

$C_2 = 14$; $n_0 = 1$

Big Ω:

$\Omega(n^3)$: $C_1 = 1/2$

$f(n) \geq \frac{1}{2} n^3$

$n_0 = \frac{10+2+1}{1-\frac{1}{2}} = 26$

Big Θ: $\frac{1}{2} n^3 \leq n^3 - 10n^2 - 2n + 1 \leq 14n^3$; $n_0 = 26$; $\Theta(n^3)$

$f(n) \in \Theta(n^3)$

3. [Marks 2]

What is the relation $\sqrt{n^n} \overset{?}{=} \left(n^{\sqrt{n}}\right)$. Is it Big-O, Big-Omega, or Θ?

Using L'Hopital:

$$\lim_{n \to \infty} \frac{\sqrt{n^n}}{n^{\sqrt{n}}} = \lim_{n \to \infty} \frac{n^{\frac{n}{2}}}{n^{n^{\frac{1}{2}}}}$$

$\frac{n}{2} - \frac{1}{2} = n-2$

$$= \frac{1}{\sqrt{x}} = \frac{\frac{n}{2} \cdot n^{n-2}}{\sqrt{n} \cdot n^{\sqrt{n}-1}}$$

$$= \lim_{n \to \infty} \frac{\frac{n^{n-1}}{2}}{n^{\sqrt{n}-\frac{1}{2}}} = \infty$$

$$\sqrt{n^n} = \Omega\left(n^{\sqrt{n}}\right)$$

1. [Marks 2]

Consider two strings T and P. Write the pseudocode to search for string P in text string T. Assuming their respective length is n and m (where $n \geq m$), what is the best and worst case complexity.

```
search (T, P){
    n ← |T| , m ← |P|
    for i ← 1 ... n-m+1 do {
        j ← 1 , k ← i
        while (j ≤ m) do {
            if (T[k] = T[j])  k ← k+1 , j ← j+1
            else if (j = m+1) return true
            else do break
    } }
    return false
}
```

Best case: $O(n)$

Worst case: $O(n-m+1) \in O(n)$

2. [Marks 3]

Suppose you are given a string S. Determine if S is palindrome. Write pseudocode to print YES if S is palindrome, and NO otherwise. A string is palindrome if it reads the same from both ends, e.g. "abbcbba", "aaa". What is the complexity?

```
palindrome (S ){
    j ← |S| , i ← 1

    while ( i ≠ j ) do {
        if S[i] = S[j] then  i ← i+1 , j ← j-1
        else do {
            print "NO"
            break
        }
    }
    if i ≥ j then
        print "YES"
}
```

$n = |S|$

Worst case: $O(\frac{n}{2}) \in O(n)$

DP version of fibonacci i

```
Procedure  fib(n)
{
    a ← 0          // f_0
    b ← 1          // f_1
    for i ← 2...n  do
    {
        C ← a+b    // f_{i-1} + f_{i-2}
        a ← b
        b ← C
    }
return C
}
```

O(n)